REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*: 10-JAN-2003

2. REPORT TYPE: Conference Proceedings, (refereed)

3. DATES COVERED *(From - To)*:

4. TITLE AND SUBTITLE: Simulator Development For Multiple Unmanned Underwater Vessels (Full Draft)

5a. CONTRACT NUMBER:

5b. GRANT NUMBER:

5c. PROGRAM ELEMENT NUMBER: 62435N

5d. PROJECT NUMBER:

5e. TASK NUMBER:

5f. WORK UNIT NUMBER: 74-6636-03

6. AUTHOR(S): MARVIN W ROE   BRIAN S BOURGEOIS   PATRICK MCDOWELL

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES):
Naval Research Laboratory
Marine Geoscience Division
Stennis Space Center, MS 39529-5004

8. REPORTING ORGANIZATION REPORT NUMBER: NRL/PP/7440--03-1002

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES):
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

10. SPONSOR/MONITOR'S ACRONYM(S): ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S):

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release,distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper is about the development of a system that will simulate the operation of multiple Unmanned Underwater Vessels (UUV's). The simulator is being designed to support the research efforts of the Position. Navigation and Timing (PNT) team of the Naval Research Laboratory (NRL), located at Stennis Space Center. In this paper, we will discuss the functionality and architecture of a simulator that will support this research. Our approach is to use a network of PC's with a vessel simulation running on each PC. An additional PC will host our Central Simulation Processes that will display simulation progress and serve as a central control for shared data and communications. This calls for the ability to create a flexible distributed real-time system that can synchronize vessel interaction in a team setting. Various combinations of simulated and physical vessel types must he allowed to support the different team member roles and our phased development approach. We present a detailed description of the architecture proposed for our simulator and discuss its operation. Finally, we will present our observation of the performance of a prototype implementation and discuss our future plans for development and testing.

15. SUBJECT TERMS

Unmanned Underwater Vessel, Autonomous, Simulation, Navigation, Position, Timing

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(Include area code)*
unclassified	unclassified	unclassified	Unlimited	15	Marvin Roe / 228-688-4937

Standard Form 298 (Rev. 8/98)

20030416 338

Simulator Development For Multiple Unmanned Underwater Vessels
(Full Draft)

Marvin Roe
Naval Research Laboratory
Stennis Space Center, MS
39529-5004, USA
228.688.3947
mroe@nrlssc.navy.mil

Brian Bourgeois
Naval Research Laboratory
Stennis Space Center, MS
39529-5004, USA
228.688.5321
bsb@nrlssc.navy.mil

Patrick McDowell
Naval Research Laboratory
Stennis Space Center, MS
39529-5004, USA
228.688.4823
pmcdowell@nrlssc.navy.mil

ABSTRACT
This paper is about the development of a system that will simulate the operation of multiple Unmanned Underwater Vessels (UUV's). The simulator is being designed to support the research efforts of the Position, Navigation and Timing (PNT) team of the Naval Research Laboratory (NRL), located at Stennis Space Center. In this paper, we will discuss the functionality and architecture of a simulator that will support this research. Our approach is to use a network of PC's with a vessel simulation running on each PC. An additional PC will host our Central Simulation Processes that will display simulation progress and serve as a central control for shared data and communications. This calls for the ability to create a flexible distributed real-time system that can synchronize vessel interaction in a team setting. Various combinations of simulated and physical vessel types must be allowed to support the different team member roles and our phased development approach. We present a detailed description of the architecture proposed for our simulator and discuss its operation. Finally, we will present our observation of the performance of a prototype implementation and discuss our future plans for development and testing.

Categories and Subject Descriptors
Distributed Real-time Systems, Autonomy, Simulation, Underwater Vessels

General Terms
Performance, Design, Reliability, Experimentation, Human Factors, Standardization, Theory

Keywords
Unmanned Underwater Vessel, Autonomous, Simulation, Navigation, Position, Timing, Communication, Acoustic

1. INTRODUCTION
The Position, Navigation and Timing (PNT) Team at the Naval Research Laboratory (NRL) is doing research to develop the necessary communication, intelligence, and multi-vessel navigation systems to support Unmanned Underwater Vessel (UUV) team operations. UUV teams have the potential to perform military and commercial survey operations of near-shore and other underwater environments [2, 3, 4]. Figure 1 presents an example of a notional UUV task force arrangement that shows how multiple vessels could work together to carry out survey operations.

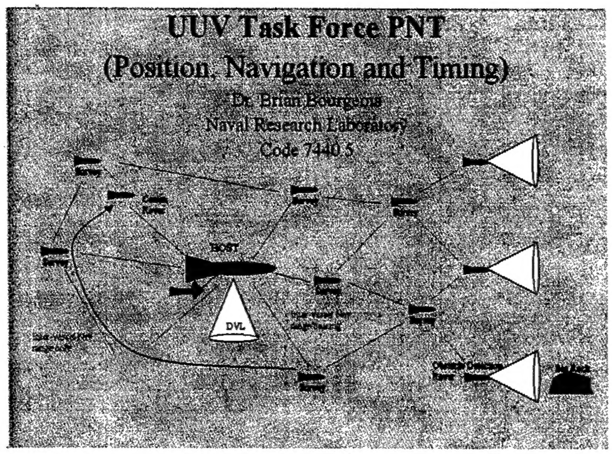

Figure 1. Example UUV task force configuration.

Figure 1 shows multiple UUV's executing a survey. The formation consists of a 'host' UUV that has been configured with a precise high-cost position, navigation and timing system. The host UUV is capable of independent operation for several days and will serve to provide position reference and communication control for the task force. Survey vessels for collecting data will be arranged in various patterns depending on the nature of the mission. High volume data transfers will be handled by a communications rover that uses short-range high bandwidth communications. The task force will also have specialized UUV's that will serve to perform obstacle detection and avoidance support. These vessels are located in the front of the formation and will notify the rest of the task force of obstacles to be avoided. Vessels in the task force that are not equipped with the sophisticated position systems will obtain range and bearing information to the other vessels via acoustic communications. By creating a team of UUV's, we can effectively increase productivity through shared resources and increased capabilities. In the underwater environment acoustic systems are the primary means for communications and for knowing a vessel's position relative to others within a team. But these systems offer only limited bandwidths resulting in significant delays for sharing information between vessels and for obtaining relative positions. This presents a challenge to navigation of vessels operating as a team and high levels of autonomy will be needed. This research will explore approaches to communications, positioning, navigation, timing and autonomy required to enable a UUV team. Determining the sensitivity of these schemes to available bandwidth, number of vessels, formation size, etc. is a key function of the simulator being developed. In the next section the simulator functionality required to support this research effort is described. In section 2 we discusses our research approach and the simulator functionality required to support it. In section 3 we present our conceptual architecture design to support the research approach and functionality requirements. Section 4 presents the goals, functionality, construction and execution of a prototype implementation in LabVIEW. Finally, in section 5 we present a summary and discuss future plans for development of our simulator.

2. SIMULATOR FUNCTIONALITY

The approach taken in this research will begin by observing a human in the loop where vessel state knowledge is displayed graphically and the human pilots the vessel. The results of these observations will be used to develop unmanned capabilities through the development of intelligent, goal based, routines that will replace the human pilots. This section addresses the functionality that will be built into the simulator to support this effort.

2.1 Goals

The system that is developed must support easy implementation of conceptual vessel sensors, positioning and navigation algorithms. An architecture that allows for plug-in components will provide the most benefit. This plug-in capability should include the ability to change the number of vessels in a simulation whether simulated or physical. Additional abilities are ease of adding or removing sensor capabilities, environment simulation modules, inter-vessel communications protocols, and autonomous control modules. The system should also provide an easy migration of algorithms such as position and navigation from simulation to physical systems. To facilitate this goal, the simulator must be able to coordinate the operation of any combination of simulated and physical vessels simultaneously as show in Figure 2.

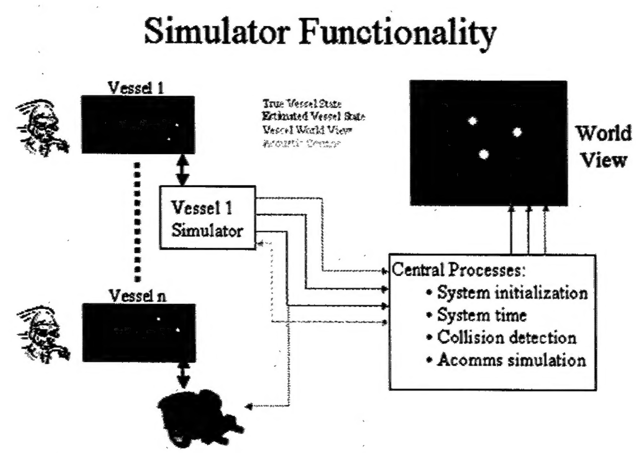

Figure 2. Simulated Vessel and Robot Control Capabilities.

The simulator functionality shown in Figure 2 shows the need to be able to control n vessels whether they are simulated, physical or a combination of the two in the same simulation. Data logging must also be provided by the system to support measurement of system performance and playback capabilities.

2.3 Approach

The simplest definition of what we aim to accomplish is to create a team of vessels that can maneuver through the underwater environment while collecting data without the need for human control. Because of the difficulties presented by working in the underwater environment and the complexities of autonomy and cooperative autonomous systems, we

have chosen our incremental approach that will allow for a smooth migration from a simulated to a physical implementation. Since the resultant system will most likely have custom sensor packages that emerge as a result of our research, we will also be able to study new sensor configurations as they migrate from conceptual simulation to land based testing and to their final implementation in the underwater environment. What follows is a generalized discussion of the integral high level components and how we plan to progress from human control in a simulated environment to autonomous control in a physical environment.

2.4 Component Concepts

The highest-level view of our approach reveals the idea of controlling vessels in an environment. For our research we identify two types of control namely human and autonomous. Furthermore, we identify the idea of a simulated environment and a physical environment. Throughout the research and development process, we will experiment with various combinations of these aspects. We may want to work with arrangements where only pure combinations of these aspects come into play such as Human Control of a vessel in a Simulated Environment. We will also need the ability to work with hybrid versions such as when we require human control of an environment that includes some simulated and some physical components. Table 1 shows the combinations that will be used throughout our development. In the Table 1, each column shows which aspects of control and environment are being used within any particular configuration of the Operational System. Each aspect configuration has a unique identifying index for reference throughout this paper. In Table 1, column 1 identifies the initial step that combines human control in a simulated environment and column 9 identifies the goal of achieving autonomous control in the physical environment. Columns 2 through 8 show configurations that may or may not be used during the development of various modules.

Table 1. Operational System Configurations.

	Operational System								
Development Phase	Init	Optional and Recurrent							End
Aspect Configuration	1	2	3	4	5	6	7	8	9
Human Control	X	X	X	X	X	X			
Autonomous Control				X	X	X	X	X	X
Simulated Environment	X	X		X	X		X	X	
Physical Environment		X	X		X	X		X	X

We could build a system for each scenario described in Table 1. Unfortunately, we would incur the overhead of developing and maintaining multiple projects and have less ability to compare results between systems. New modules would have to be modified to interact with each system as they progress from the simulated to the physical implementation and re-modified if there becomes a need to go back to a previous level for further development or testing. A good design would allow any given module the

ability to return to a previous Aspect Configuration of the Operational System if necessary. A simulator that has a great ability to blend combinations of these high level concepts will produce greater success and productivity while arriving at the final system.

3. SIMULATOR ARCHITECTURE

In this section we will consider our ideas of control and environment and how we expand them into a high level conceptual architecture.

3.1Vessel Concept

To further explore our architecture we need to introduce a high-level concept for a vessel. A vessel consists of sensors, actuators, and a control system that manages them as they interact with the environment as shown in Figure 2.

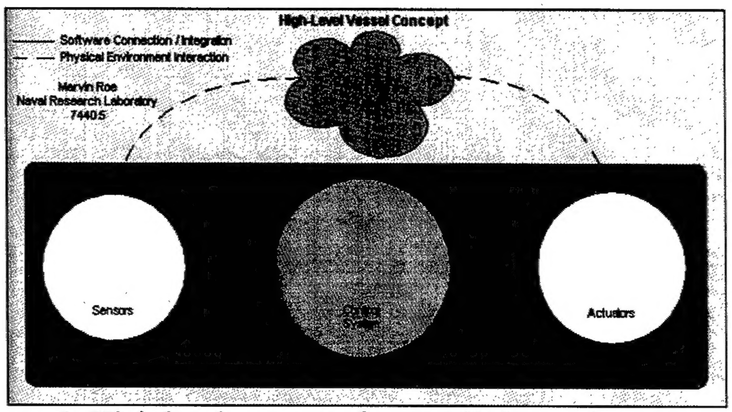

Figure 2. High-level concept of a vessel's internal components.

Sensors are viewed as devices that act as input from the environment such as speed and bearing and as input components of communication systems. The Actuators are viewed as devices that act as end effectors, propulsion systems that provide maneuverability and output components of communication systems. The Sensors and Actuators modules will provide a common interface simulated to sensor and actuator hardware drivers when needed and provide a place to perform functions relative to environmental simulation.

3.2 High-Level Concept

Considering our control and environmental aspects together with our conceptual vessel, we can arrange our ideas into a diagram that shows an architectural relationship that will support any of the desired combinations illustrated in Table 1 and discussed in Section 2.4.

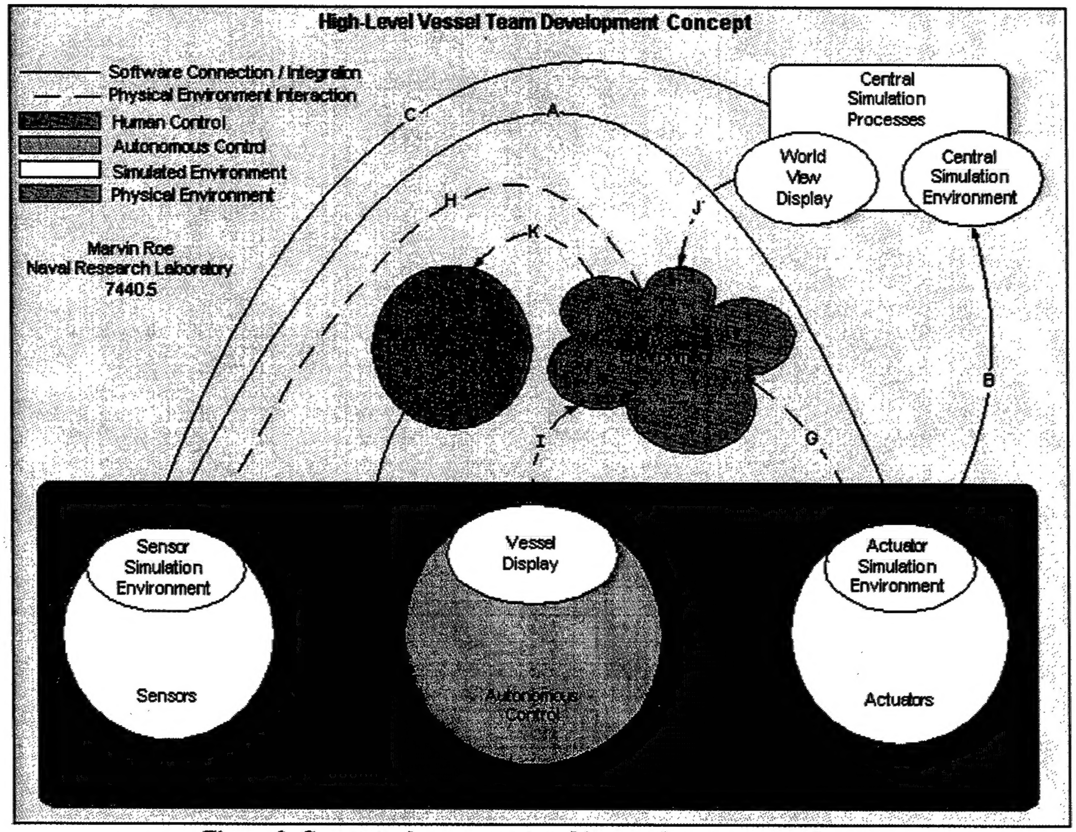

Figure 3. Conceptual component architectural arrangement for one Vessel and its connection to the Central Simulation Processes.

Figure 3 is an illustration of how the pieces will fit together. To explain how this architecture will work, we intend to provide a general description of the component parts and their arrangement followed by the presentation of specific examples that will illustrate operational system configurations that will be based on this architectural design.

The first two examples will explore implementations that represent the end points of the flexibility of the architecture. The two scenarios are Human Control in a Simulated Environment and Autonomous Control in a Physical Environment. These will be followed by general examples that will illustrate the stages of system migration between the two endpoints. Finally, we will discuss how this architecture will support research and development currently being done by the PNT team.

3.2.1 Component Overview

Prior to exploring specific examples, we give a general description of the diagram components in Figure 3. This diagram shows the high level components for one vessel and its connection to the physical and simulated environments. Additional vessels would have their own connections to and share the physical and central simulated environments. The solid lines A through F that connect various components in the diagram represent software connections or integration. This means that components connected by solid lines can be closely integrated or reside on separate machines. The dashed lines G through K are included to complete the conceptual idea as robots, screen displays, and operators interact through the physical environment. Depending on the configuration of

the operational system, modules can perform processing or act as a pass through to complete the system circuit. The Sensors module provides a common API to the Autonomous Control module. Drivers for Sensors whether simulated or real will be added here. This module also contains the Sensor Simulation Environment module. The Sensor Simulation Environment Module provides a place to insert any extra processing that is necessary to produce the desired environmental affects on sensor values. The Autonomous Control module receives input from Human Control, provides the Vessel Display and performs the major processing functions for a given vessel. The Actuators module provides a common API to the Autonomous Control module. Drivers for propulsion systems and any other end effectors whether simulated or real will be added here. This module also contains the Actuator Simulation Environment module. The Actuator Simulation Environment Module provides a place to insert any extra processing that is necessary to produce the desired actuator effects on environmental values. The Human Control module represents input from a joystick, keyboard or other input device into the Autonomous Control module. The Central Simulation Processes contains all of the necessary support for team interaction of multiple vessels such as shared state values. It contains the Central Simulation Environment and World View Display modules. The Central Simulation Environment module provides a place to insert any extra processing that is necessary to produce the desired environmental effects between actuators on different vessels. The World View Display module will provide a graphical representation of each vessel's true position and heading and each vessel's best guess of its position. The Physical Environment cloud represents the interaction of actuators and sensors in the real world. It provides a way to note the relationship between effects of the actuators of a robot or UUV or computer screen and what the sensors or operator 'sees' as a result. It's important to recognize that the lines of the conceptual diagram only represent high-level relationships. They don't necessarily depict direction of data flow or timing of data flow. They also don't show any type of timing or synchronization between the component parts or the system as a whole during execution. This would be evident if we look at the idea of an operator controlling a land-based robot while viewing its position on the Vessel display. From the Physical Environment point of view the robot would be moving, the vessel display updating, the operator controlling the joystick and viewing the results by observing the robot and or the vessel display all at the same time. All of the same holds true for the other component pieces of the architecture. The research being done is not limited to determining how various tasks will be accomplished but also the synchronous and/or asynchronous execution of these tasks that best provides a solution.

3.2.2 Human Control in a Simulated Environment

This example is based on Aspect Configuration 1 in Table 1 and will be the explanation of a single vessel under Human Control in a Simulated Environment. Starting with the Human Control bubble in the diagram, this represents the operator's use of some type of input device such as a keyboard or joystick that will translate control signals that ultimately drive the vessel. Path D in the diagram represents the software connection or integration of the input device to the Autonomous Control structure. For this example, the Autonomous Control structure would act as a pass through or basic conversion of the input signals to values or commands such steering or speed changes that can be sent to the Actuators. Once any necessary conversion has been done, the information would

next be sent to the Actuators structure via path F and directly into the Actuator Simulation Environment bubble where the necessary calculations would be done to determine just what the sensors would see as a result if an actual propulsion system existed. For this example, we will say that our vessel has a compass and GPS system. This means that the Actuator Simulation Environment would have to calculate the resultant compass heading and GPS coordinates so that they could be passed via path A to the Sensor Simulation Environment bubble. Since the simulated values were calculated on the Actuator side, for this example the Sensor Simulation Environment bubble would be a pass through with no changes. Next the sensor values would pass through path E and into the Autonomous Control structure where the values can be used to display the vessels current position and heading. To complete the conceptual connection, the operator will have the updated position and heading as feedback on their performance as they view the Vessel Display depicted as interaction through the Physical Environment via path I and finally through path K.

3.2.3 Autonomous Control in a Physical Environment

This example is based on Aspect Configuration 9 in Table 1 and will be the explanation of a single vessel under Autonomous Control in a Physical Environment. As an added note, the vessel is a land-based robot with two drive wheels and a balance caster and the physical environment is our robotics laboratory. Starting with the Autonomous Control bubble in the diagram, evaluation of Sensor information acquired via path E will be used to determine what commands will be sent to the Actuators via path F. As the commands are processed the drive wheels in this case will affect the robots interaction with the physical environment through path G. Sensors such as encoders or a compass will sense the effects of the actuators actions through path H in the diagram and will be able to make this updated information available to the Autonomous Control structure for further analysis. Both the Actuator Simulation Environment and the Sensor Simulation Environment modules would not perform any modification related to simulating the environment.

3.2.4 Observations

This high level architecture will support both ends of the spectrum for our system configuration. The substitution of a UUV with the appropriate autonomous control and sensor and actuator systems for the land-based robot in the example given in section 3.2.3 would complete the concept. It should be noted here that the Operational Systems shown by Aspect Configuration 3 and Aspect Configuration 7 in Table 1 could be derived from the examples given in Sections 3.2.2 and 3.2.3.

3.3 Intermediate Operational Systems

Now that we have explored the two end points of the flexibility of this design, we will explore several more examples in order to illustrate more of the functionality that this architecture provides.

3.3.1 Autonomous Control in a Combined Simulated and Physical Environment

This example will expand on the one given in Section 3.2.3 and will illustrate how the architecture will accommodate an Operational System with Aspect Configuration 8 from Table 1. At the end of that example discussion we stated that both the Actuator

Simulation Environment and the Sensor Simulation Environment modules would not perform any modification related to simulating the environment. If we first look at the Actuator Simulation Environment module, we have the opportunity to interject a more realistic movement of the land-based robot since our main focus at studying the maneuverability of underwater vessels. We can intercept the commands coming in from the Autonomous Control and affect things like how fast the vessel accelerates, turns or even stops. An example would be when a command to set the speed from 5 knots to 0 knots is received. Instead of the robot stopping suddenly, we can modify this to gradually slow the robot until it finally stops in a more realistic fashion. As for the Sensor Simulation Environment module, we can modify things like the encoder values that would normally tell us how far the robot has traveled and scale them so that we can study the vessels performance over a larger virtual area. The added benefit illustrated by this example is that we can effectively changed the Operational Configuration without changing any of the commands or processing between the Sensors, Autonomous Control and the Actuators processes.

3.3.2 Human / Autonomous Control and Simulated / Physical Environment

This example will expand on the one given in section 3.3.1 and will illustrate how the architecture will accommodate an Operational System with Aspect Configuration 5 from Table 1. An example application of this configuration would be a system that supported a set of high-level commands that could be issued from a Human Control point of view. The Autonomous Control would evaluate high-level commands signaled by the Human Control through path D. A decision would then be made by the vessel's Autonomous Control process as to how it would go about executing the request. A valid command may involve the Human Control telling the vessel to go from where it is to a designated point. The Autonomous Control process could carry out the request deciding for itself what is the best path to take and also handle avoiding unforeseen obstacles along the way.

3.3.3 Cooperative Autonomous Operations

To this point, the examples have applied to single vessel configurations. The final components are utilized when we want multiple vessels to work together. When considering the idea of multiple vessels interacting with each other, the need for connectivity between them arises. The path that provides this ability leaves the vessel through its Actuator Simulation Environment module and travels through path B to the Central Simulation Environment. The return path is from the Central Simulation Processes through path C and back into the vessel through its Sensor Simulation Environment. From this high-level view of the CSP connection for a given vessel, there are two main types of communication occurring throughout each simulation run, namely Operational System and Simulated communications.

3.3.3.1 Operational System Communication

The first type of communication is for supporting the Operational System. This is a software level communication that facilitates vessels initializing their connection to the CSP and thereby becoming a part of the team. This type of communication will also be used by vessels to transmit the current value of their 'True State' and to request the 'True State's of other vessels running in the simulation. These 'True States' represent the actual state of a vessel in regards to position, speed and bearing. These values can be fed

into learning routines as part of their development. Also provided by this operational communication will be access to timing information such as a simulation heartbeat and/or synchronization to a master clock. This communication effectively allows a vessel to become part of the simulation and handles Operational System related issues.

3.3.3.2 Simulated Communication

A large part of the research to be done surrounds issues related to improving underwater communication systems. The study is focused on all aspects including the hardware, the protocols (software) that will utilize the chosen hardware and the data that will actually be communicated through the system. This means that we can effectively split the idea of simulated communication into three sub-types. In short, to send and receive a simulated communication message, the message data will be converted to the simulated protocol and sent to the Central Simulation Environment using Operational System Communications as discussed in Section 3.3.3.1. There it will be distributed to those that will hear it based on the hardware and environment being simulated. This will be such that the vessel that receives the message will only recognize it at the precise time it would have sensed the communication if it was actually in the target physical environment. Once officially received, the message would be pulled back out of the simulation protocol and be converted to message data to be used by the target vessel(s). This type of message data passing is ultimately how vessels will acquire 'Estimated State' and other information from other vessels. The 'Estimated State' of other vessels, contains mostly the same data as the 'True State' data discussed in Section 3.3.3.1. The difference is that the 'Estimated State' is each vessel's best guess as to what their position and heading are and other related data. While not accurate like 'True State' information, 'Estimated State' information is still useful to learning processes and will be the only feedback for team success when performing Autonomous Control in the Physical Environment.

3.3.3.3 General Cooperative Support

The CSP is the main set of processes that will coordinate the interaction of multiple vessels. The CSP will essentially maintain multiple connections to multiple vessels in order to facilitate the necessary connectivity. One of the CSP components is the World View Display. The World View Display will use the transmitted 'True' and 'Estimated' states of all the vessels in the simulation to show a graphical representation of each vessels true and estimated position, bearing, and position history. The information displayed on the World View will provide feed back of team performance during operations. In addition, the CSP will include a Central Simulation Environment module that will facilitate any processing for group related simulation. Finally, the CSP will also coordinate each simulation by handling tasks such as the entry of new vessels into the simulation.

3.3.4 Observations

We have discussed how this high level architecture will support the transitional stages between the end points of our system development. It should be noted here that the Operational Systems shown by Aspect Configuration 4 and 6 in Table 1 could be derived from the example given in Section 4.3.2. Furthermore, Component Configuration 2 can be derived from the examples given in Section 4.3.1 and Section 4.2.2. In the next

section, we will discuss examples that show how this architecture will support research already being done by the PNT team.

3.4 Research Support Examples

The following two examples are provided to show how this architecture will support current research in the area of UUV team operations. These examples show the advantages that can be gained by adopting a common architecture that will support development from simulation to end system.

3.4.1 Sensor Configuration and Use

Current research surrounding the study of UUV team formation during various phases of a mission has been addressed in [1]. This ongoing study is approaching control mechanisms from a biological perspective instead of comparing position information for the purpose of entering into and maintaining certain formations. The general idea is based on having leader agents emitting a tone from a speaker that follower agents listen to with a right and a left microphone. The microphone information is then passed through a neural network that produces the next heading and speed values necessary to maintain the desired formation. Successful software simulation of the theory has been performed and is now in the process of being implemented on land-based robots for further evaluation. The architecture presented in Figure 3 would support the development of this theory in all stages from software simulation to physical implementation. Few changes would be necessary to move from software-only simulation to experimentation on the robots. After moving the software to the robots, the necessary drivers would have to be integrated to complete the migration from a Simulated Environment to a combination Simulated/Physical Environment. The Sensors module would have to be updated to integrate the left and right microphones and heading and speed information from the robot. Finally, the Actuators module would have to be updated to include control of the speaker and robot drive mechanisms. A third and final configuration would be one that is configured to work with a UUV and it's drive control system and hydrophone configuration. Once initially created, these configurations could be switched between and compared much more closely while development continues until migration from the simulated to the physical world is complete.

3.4.2 UUV Task Force Configuration

Due to cost, payload and computational requirements of the necessary systems to perform required tasks to successfully implement the team solution, we plan to incorporate the use of several differently configured UUV's in the same operational exercise as discussed in [2], shown in Figure 1, and briefly discussed in Section 1. The idea is that a few larger and more expensive UUV's with more sophisticated navigation and positioning systems could be used to manage the operations of a team of cheaper more maneuverable UUV's. For simulation purposes in software and on land-based robots, the physical movement and maneuverability would have to be configured to take this into account. By modifying the Actuator Simulation Environment and Sensor Simulation Environment modules for each UUV, we can achieve the necessary maneuvering responses for each type of vessel. This architecture would allow for a common place allow for software-only simulation and land-based simulation of the movement of UUV's.

3.5 Concept Review

This architectural design will support both the development of internal functionality of the various vessel types that will work together as a team and the development of the communication, timing and navigation schemes that will enable them to work successfully together. This high-level architecture will transcend all phases of research and development. As long as any piece can operate within the same high-level architecture we can move seamlessly from human control in a simulated environment to autonomous control of the physical environment. This system design will not only aid in development but also improve the maintainability and configuration of the end system. Configuring such a system for different missions requires changing such things as navigation schemes for different surveys in different types of ocean environments. This system will allow for easier implementation of improvements in the future as the technology grows. As this architecture is still in the development stages and design and implementation issues are researched we will build smaller prototypes as necessary to prove basic concepts and illuminate unforeseen issues while designing the overall operational simulator system. In the next section, we will explore a prototype that was built in order to help explore the advantages of using LabVIEW for parts of the system and to exercise the use of our lab equipment to date.

4. INITIAL IMPLEMENTATION

As part of the architecture and software design process, many smaller more focused prototypes will be built to test programming languages and their interoperability, test our lab equipment such as wireless Ethernet and robots and to provide intermediate research and study capabilities until the final system is complete. In this section we will discuss a prototype that was created to show the integration necessary to control our robots while perform tasks using a wireless Ethernet connection to dedicated desktop machines.

4.1 Functionality

The robot control system was designed to prove the functionality; control the Robots using their low level API calls from remote desktop machines, obtain position and heading information from each robot and compare to known values, allow operator control via a joystick, evaluate the use of LabVIEW socket programming to communicate commands and return data over the wireless Ethernet, provide session logging and playback, to provide a graphical interface that shows the position of each robot to the user.

4.2 Construction

The PNT team configured each robot with Windows 2000, wireless Ethernet capability and LabVIEW. Next, the robot API's were wrapped inside a Dynamic Link Library (dll) so LabVIEW could easily access them. A second dll was created so LabVIEW obtain input from the joystick. LabVIEW clients and servers were then developed to provide an interface for the operators and connectivity to the robots. The floor of the robot lab was then used to layout a 12-foot square grid that consisted of 4 concentric squares and an overall cross hair in the middle as shown in Figure 4. The letters T, B, L and R were placed as needed to indicate Top, Bottom, Left and Right respectively. This grid design was replicated to scale within the operator GUI in LabVIEW and provides icons to

represent each robot's position on the floor. The GUI also allows for the input of an offset for the x and y values since the encoder values for each robot start off at 0,0.

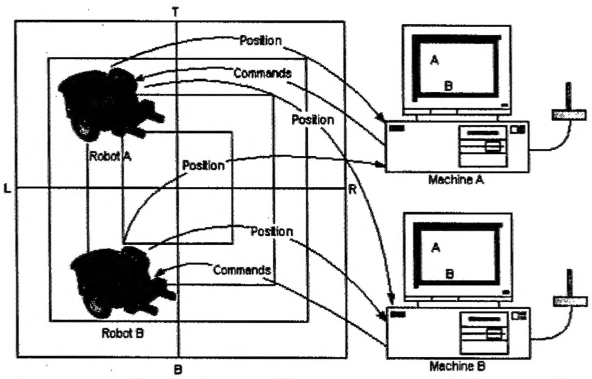

Figure 4. Overview of Prototype and Robot Lab Setup.

4.3 Execution

For each system run we first place each robot on the floor. We usually place one directly in the center so no offset has to be entered and place another on a corner of one of the concentric squares so we can easily enter it's offset. The robots are initialized and the desktop machines are initialized along with the LabVIEW server and client applications. Operators can command their robot to move forward by pushing the joystick forward and back for reverse. Turning each robot is done by tilting the joystick either to the right or the left. When the joystick is returned to its resting position the robot stops at its current position. While the robots are being maneuvered around the grid, they are sending position and heading information to each operator to be viewed in the GUI as shown in Figure 5.

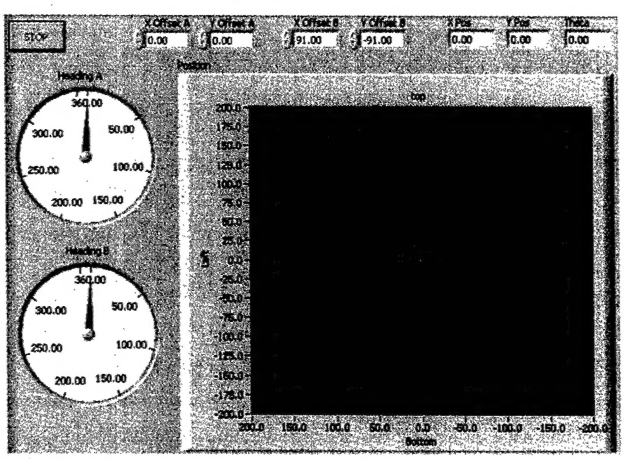
Figure 5. Operator GUI.

Figure 5 shows the GUI that shows each operator where the robots are, the white icon in the middle of the grid and the green icon on the top right corner of the second square from the middle. The red lines represent the grid lines that are on the floor and the top bottom and left have been labeled on the screen. The GUI also provides heading information in the form of the compass dials on the left side of the GUI. We now have

the ability to observe the true state, or position and heading, of the land based robots verses the 'estimated state' that they are reporting to each operator. The final GUI provides control of logging and playback capabilities as shown in Figure 6.

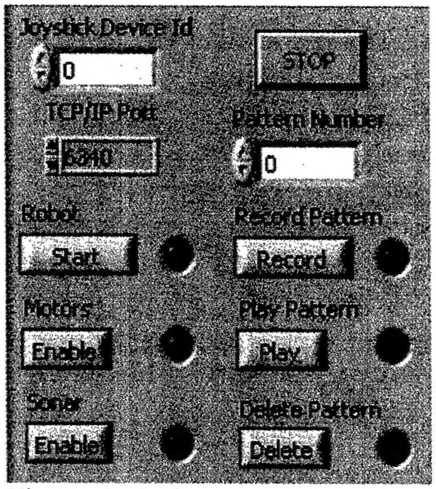

Figure 6. Robot Control and Logging and Playback.

Figure 6 shows the main control panel for initializing each robot and for recording and playing back session operations. To record session operations, the user selects a number and then clicks the Record button. While the Record button is on, the associated robot stores motion commands to a file on the robot's hard drive. Any particular session, or pattern number, can be played back by selecting the number and then pressing the Play button. We added this feature so we could study the accuracy of our robots ability to reproduce certain patterns exactly as they had done previously and to show the ability to log commands and play them back.

5. FUTURE WORK AND SUMMARY

Future work for this project will include programming language and tool evaluations, progression into lower level concept and design specification, and the production of prototypes to test solutions to challenges as they arise. Performance and functionality comparisons will be conducted between LabVIEW, Java and CORBA tools to see which tool or combination of tools will best suite our needs. Further evaluation of the architecture will be conducted in order to define specifications that will support distributed real-time systems, pug-in capabilities and platform flexibility. The task of creating a team of UUV's is complex and will require the work of a team of researchers to achieve. The simulator must provide the functionality presented in this paper so that researchers can successfully combine and evaluate their efforts.

6. ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research through the Naval Research Laboratory under Program Element 62435N. The mention of commercial products or the use of company names does not in any way imply endorsement by the U.S. Navy. Approved for public release; distribution is unlimited. NRL contribution number NRL\PP\7440 - - 03-1002.

7. REFERENCES

[1] P. McDowell, J. Chen, B. Bourgeois, ' UUV Teams, Control from a Biological Perspective', *Proceedings of the Oceans 2002 Conference*, Biloxi, MS, 29-31OCT02

[2] B. Bourgeois and P. McDowell, "UUV Teams for Deep Water Operations", Underwater Intervention 2002, New Orleans, LA February 27 – March 2, 2002.

[3] R. Wernli "Low-Cost UUVs for Military Applications", *Sea Technology* December 2002

[4] T. Curtin and J. Bellingham "Guest Editorial – Autonomous Ocean Sampling Network", IEEE Journal of Oceanic Engineering, vol.26, no. 4, October 2001